Intell Ind Syst (2016) 2:303–318
DOI 10.1007/s40903-016-0059-4

ORIGINAL PAPER

Enabling Residential Demand Response Applications with a ZigBee-Based Load Controller System

Avijit Saha[1] · Murat Kuzlu[1] · Manisa Pipattanasomporn[1] · Saifur Rahman[1]

Received: 2 November 2015 / Revised: 31 August 2016 / Accepted: 3 September 2016 / Published online: 21 September 2016
© Springer Science+Business Media Singapore 2016

Abstract Home energy management (HEM) systems are getting increasingly popular in both academic and commercial communities. Most HEM solutions rely on either smart appliances or plugging traditional appliances into smart outlet/smart plugs which can communicate with the central HEM server. Conventional non-communicating power-intensive loads that are directly hard-wired to the supply panel of the house in the U.S., e.g., water heaters and clothes dryers, therefore are difficult to be made part of an HEM system. This paper describes the design and implementation of a microcontroller and ZigBee based load controller system that allows these loads to be controlled by an HEM system. The proposed system consists of four load controllers (LCs) that monitor and control four power-intensive household loads, and one communication controller (CC) that acts as a communication medium between LCs and a central HEM unit. The CC and all the LCs are packaged into one load controller box that can be placed beside the main circuit panel of the house and the wirings to the power-intensive loads are routed through the LCs. Within the box, the LCs and the CC communicate via hard-wired serial peripheral interface. The CC then communicates with the HEM system via ZigBee. The proposed system is expected to offer a cost-effective method to enable traditional hard-wired power intensive loads to participate in an HEM system.

Keywords Load controllers · Power intensive loads · Home energy management · ZigBee

Introduction

As a part of the smart grid architecture, a home/building energy management (HEM/BEM) system can be an effective tool to reduce peak electricity demand and improve electric utility load factor [1]. It can help mitigate power system stress conditions and improve the utilization of generation, transmission and distribution resources. As a result, there have been a lot of interest in developing effective HEM/BEM algorithms [2–8], as well as hardware infrastructures/ components for complete HEM/BEM solutions [9–17]. For communication and control of household loads, most HEM solutions rely either on an existing network of wireless sensors [9] or new design of smart outlets/smart plugs/smart sensors in a system that may use ZigBee [10], power line communication (PLC) [11], ZigBee and PLC [12], infrared (IR) [13], BACnet [14], or cloud-based technologies [15]. Besides academic literature, there exist commercial HEM products [18–20]. Most of these products are only compatible with selected proprietary smart plug products and do not follow any open standards. At the same time, there are a number of off-the-shelf smart plug products available in the market that can be used with some HEM systems [21–24].

As an automated home/building energy management system must be able to communicate with appliances, it is necessary to either replace old non-communicating loads with modern communicating appliances or make their control possible through the use of smart outlets/plugs. None of

✉ Avijit Saha
avijit@vt.edu

Murat Kuzlu
mkuzlu@vt.edu

Manisa Pipattanasomporn
mpipatta@vt.edu

Saifur Rahman
srahman@vt.edu

[1] Advanced Research Institute, Virginia Tech, Arlington, VA, USA

these options are feasible/cost-effective for traditional power intensive loads that are directly hard-wired to the main circuit panel of an old-style house (i.e., water heater, clothes dryer, etc.). Although wireless smart plugs provide the convenience of plug and play control, they can only be used for plug loads that are connected to power sockets/outlets and cannot be used with traditional hard-wired loads without major changes in the building wiring system. Also, most of the existing smart plug solutions can only control 120 V loads with current rating limited to a maximum of 15 A. This limits them from being used with 240 V power intensive loads in the U.S., like electric clothes dryers, electric water heaters, and level-2 electric vehicle chargers, which require 20–30 A breakers. Another issue with most commercial smart plug products is that they only provide ON/OFF control and ON/OFF status information, but cannot provide power consumption information, which is required for decision making in any HEM/BEM.

There also exist solutions provided by utilities/demand aggregators to control traditional power-intensive loads during peak demand periods. For example, in eastern US, Dominion Virginia Power offers a 'Smart Cooling Rewards' program [25], in which an AC cycling switch is installed at the outdoor cooling system of a house, and a signal to this switch can be used to cut the cooling load of the house during a peak demand period event, in exchange of a one-time bill credit to the customer in each year of participation. Rappahannock Electric Cooperative offers a similar concept for water heater control [26], where a communicating switch is installed on the water heater of a house for enabling the control of the water heater load through radio/power line signal. These solutions give a utility the ability to control only one particular load in the house during the time of need, which may not be convenient for the customer at that certain time. Therefore, a better approach is to design a solution that enables power-intensive loads in a house to be controlled based on customer priorities and preference settings. The hardware solution proposed in this paper is designed in this context.

This paper discusses the architecture, design and implementation of a microcontroller and ZigBee [27] based load controller system that can enable non-communicating, non-plug-and-play, power intensive loads to communicate with and be controlled by an HEM system. The design presented here shows four load controllers for enabling the simultaneous control of four power-intensive loads, but the design is scalable and can include a higher number of loads. The system offers a robust, cost-effective, and easy-to-install solution to the mostly ignored issue of including these power guzzler loads of old houses into the HEM picture, thereby providing more opportunities for energy saving and demand response (DR). The system also offers a wide array of measurements for these loads, which can be useful for different applications besides energy management, like data trend analysis or fault detection.

The rest of the paper is organized as follows: "Proposed System Architecture" section presents the architecture of the proposed ZigBee-based load controller system. "Implementation Parts of the System" section discusses the design and implementation of system components. "Working Principle" section discusses the main working principle, and "Communication Protocols in the System" section and "Software Design of Load Controller and Monitoring/Control Systems" section elaborate on the communication protocols and software components of the system. "Implementation Results" section shows the different components of implementation along with results.

Proposed System Architecture

The proposed system contains four load controllers (LCs) that can provide measurements and control of four power-intensive loads, e.g., space heater, water heater, clothes dryer and level-2 electric vehicle charger. It also contains one communication controller (CC) that provides the link between LCs and a central HEM unit. The overall concept is shown in Fig. 1. To make the system compact, robust and easy to install, the CC and the LCs are all packaged into one enclosure box. The box can be placed beside the main circuit panel, and the wiring from the circuit panel breakers to the loads are re-routed through the LCs. Each LC in the proposed design is able to measure voltage, current, power factor, and power of the connected load and on-board temperature values, as opposed to many commercial off-the-shelf smart plugs with limited measurements. The CC communicates with LCs through hard-wired connections via Serial Peripheral Interface (SPI) protocol. The CC has a ZigBee end device router on-board to communicate wirelessly with the ZigBee coordinator of the HEM. This effectively reduces cost and complexity in the design of LCs, and utilizes only one wireless link from HEM to the controllers. The proposed design uses ZigBee communications as opposed to WiFi due to its lower power consumption and cost [28]. ZigBee also has sufficient communication range with mesh networks for residential applications.

In U.S. residences, each circuit breaker in the main circuit breaker panel can disconnect an individual 15 A or 20 A electrical circuit (served by one hot line and one neutral line), or an individual power-intensive load (served by two hot lines and one neutral line) with a typical rating of 30 A. To connect the proposed load controller box to power-intensive loads, each of which is served by two hot lines, one of the hot lines is routed from the circuit breaker through the relay of the LC. This will allow the LC to make or break the connection to the load from its power supply. Special attention is needed to

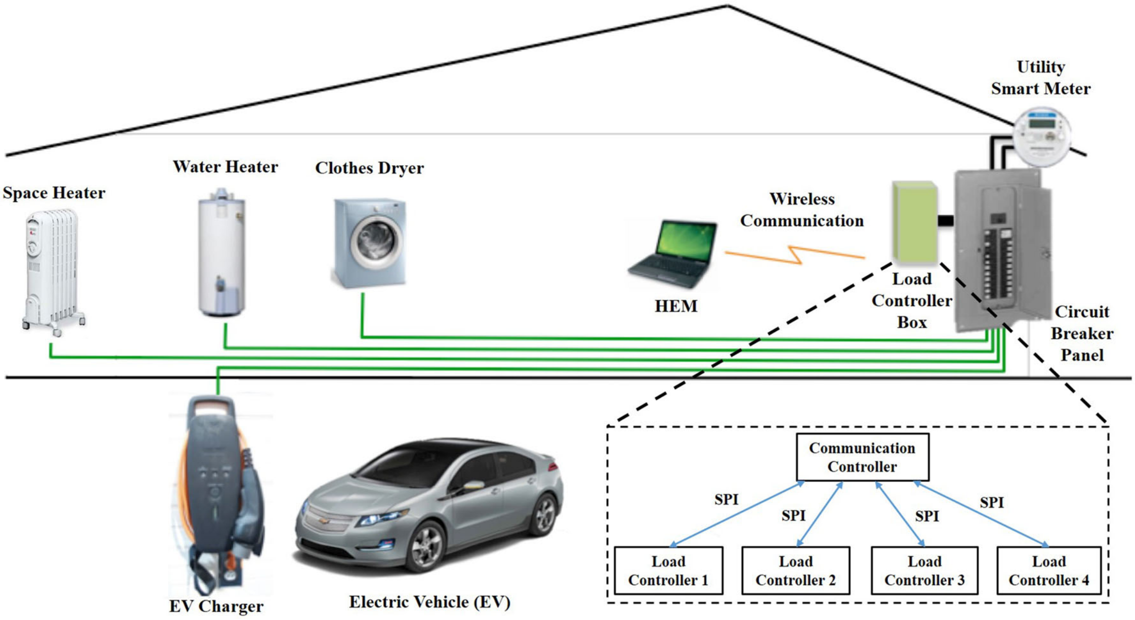

Fig. 1 Overall concept of the proposed load controller system

connect a clothes dryer to the proposed load controller box. Since only the clothes dryer's heating coils should be disconnected, the appropriate hot line should be routed through an LC relay.

Although this system is designed to have four LCs, the concept used in this design is easily scalable and can be followed to design load controller systems for a larger number of loads. Note that this system was intended to control four power-intensive appliances, including a space heater, an electric water heater, an electric clothes dryer, and a level-2 electric vehicle charger. These are resistive loads and are not sensitive to frequent power supply interruptions.

For HVAC, such frequent power supply interruptions may shorten compressor life. Therefore, the proposed load controller is not intended for HVAC control. Rather, it is recommended to control HVAC by adjusting its temperature set points through a thermostat [29].

Implementation Parts of the System

This section covers the hardware design of CC and LCs.

Communication Controller (CC)

As the CC acts as a communicating middle-man between HEM and LCs, it should be designed so that it can support both UART (Universal Asynchronous Receiver/Transmitter) (for ZigBee communication with HEM) and SPI (for communication with LCs) protocols simultaneously through different ports. Also, it needs to have an analog-to-digital converter (ADC) for temperature reading and I/O ports for the box fan and peripheral indicators. Any commercial off-the-shelf microcontroller product can be used that includes all features required for the CC design and has enough memory capacity to handle associated communication overhead. Figure 2 shows the block diagram of the hardware of the CC.

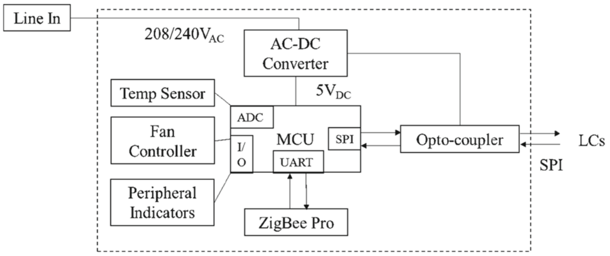

Fig. 2 Block diagram of the communication controller (CC) design

As can be seen from the figure, the micro-controller unit (MCU) is the center of the design. It controls all other components and is powered by the on-board AC–DC converter. It converts the supply 208/240 VAC to low voltage DC to power-up different components on-board. These components include a temperature sensor for measuring ambient temperature, a box fan for maintaining the temperature inside the box, the fan controller sub-circuit, some peripheral indicator LEDs, an on-board ZigBee Pro device and the MCU itself. There is a separate isolated GND (ground) AC–DC converter that powers the opto-coupler which isolates the SPI lines to LCs.

The ZigBee Pro device is connected to the UART TX/RX (transmission/reception) port of the MCU. The SPI lines through opto-coupler are connected to the SPI port's clock, MISO (Master In Slave Out), MOSI (Master Out Slave In) and chip select I/O pins of the MCU. The MCU sends and receives bytes from the HEM through UART, and from the LCs through its SPI pins. The temperature sensor is connected to the ADC port in the MCU, which samples the temperature induced voltage value produced by the sensor. The peripheral indicator LEDs are connected to different I/O pins of the MCU. The LEDs are used to indicate communication activities on UART and SPI pins and communication errors with LCs. The fan controller sub-circuit is also connected to the I/O pin of the MCU which signals fan ON/OFF status.

Load Controller (LC)

LC has less communication overhead compared to CC, but has more responsibilities concerning ADC sampling and load control through a relay. Hence, a typical industry standard microcontroller used for voltage, current, power factor sampling with SPI capabilities and some I/O functionality should be sufficient for this design. Figure 3 shows the LC design.

As shown, the MCU has separate ADC ports for simultaneous voltage and current reading functionalities. Its SPI pins are connected to the SPI lines coming from CC opto-couplers. An AC–DC converter is used to convert 208/240 VAC to 12 VDC (for operating the 12 V relay) and a voltage regulator converts 12 VDC to 3.3 VDC to power up the microcontroller and other components. A separate on-board circuitry (analog front end) is used for sensing voltage and current values appropriate for MCU ADC, which is designated as the data acquisition block. The relay driver sub-circuit and the peripheral indicator LEDs are connected to I/O pins in the MCU. The relay driver is used to drive the 12 VDC relay rated at 240 VAC, 30 A current. The relay is normally closed (NC), so that any un-foreseen incident with the hardware does not interrupt load power. Peripheral indicators, i.e., LEDs, are used to indicate SPI activity, relay status and communication errors. The MCU also has an internal temperature sensor to measure the MCU temperature.

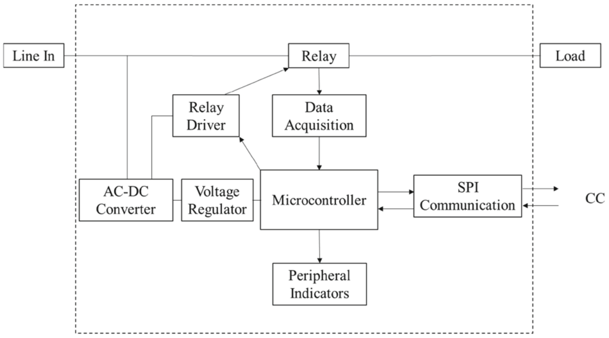

Fig. 3 Block diagram of the load controller (LC) design

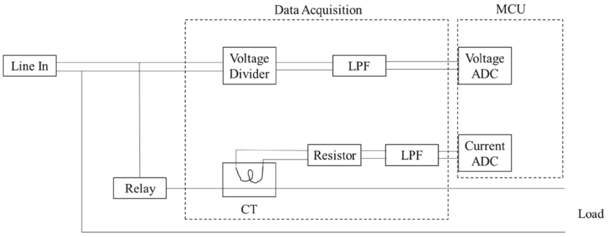

Fig. 4 Block diagram of the data acquisition in LC

Figure 4 shows details of the data acquisition circuitry. As shown in the figure, a voltage divider circuit is used to sense and convert line voltage to lower values appropriate for MCU ADC. A current transformer is used to convert line current to a lower value and then a resistor is used to convert it to voltage, which is sampled by the ADC. Low pass filters (LPFs) are used in both voltage and current lines to reduce noise in measurements. The sampled ADC values are multiplied by appropriate multiplication factors to find the actual voltage and current values. The relay in an LC is used to cut off the load from power supply.

Working Principle

Working principles for CC and LCs are discussed below:

The CC assumes the following responsibilities:

- Send a request for data to LCs and fetch power data from LCs in return via SPI.
- Package and send LC data to the HEM via ZigBee once the data request is received from the HEM.
- Send operation status (success/fail) to the HEM via ZigBee upon receiving control commands from the HEM.
- Send ON/OFF commands to LCs (based on HEM commands) and receive response from LCs via SPI.
- Measure temperature inside the enclosure via a temperature sensor, and implement box fan control based on automated decisions or commands from the HEM.

Each LC assumes the following responsibilities:

- Gather electrical data (i.e., voltage, current, power factor and frequency) from its connected load and temperature data from internal sensor in the MCU.
- Package and send data response to CC via SPI once the request for data is received.
- Acknowledge the receipt of ON/OFF command from CC via SPI.
- Control ON/OFF status of relay once a command from CC is received.

Communication protocols and software components of the proposed system are designed based on these responsibilities, as discussed in following sections.

Communication Protocols in the System

This section discusses communication protocols used in different system components.

SPI Communications Between CC and LCs

The 4-pin master-slave SPI protocol is implemented in the proposed design between CC and LCs. In this protocol, five lines are required for communication, which are clock, MISO (master input/slave output), MOSI (master output/slave input), SS (chip select of slave) and GND (reference). The clock, MISO and MOSI lines are shared by CC and LCs in a bus-type architecture. Each SS line from CC goes to one LC to allow selecting one slave for communication at a time. GND is common among all components. The clock is supplied only by the master when it needs to communicate with the slave.

ZigBee Communications Between CC and the HEM Unit

As ZigBee is asynchronous communication, no clock signal is necessary between CC and the HEM. The communication only uses one TX line for transmission, one RX line for reception, and the common GND line. The UART port of the CC is used to read and write bytes from the ZigBee end device, which can then be transmitted to the HEM's ZigBee coordinator wirelessly. The ZigBee device at CC and its coordinator at the HEM are configured to be in the same personal area network (PAN) to allow communications with each other.

Overall Communication Scheme

The CC has a timer which generates an interrupt at 1-s intervals. Based on the timer interrupt, the CC issues data requests to all LCs by communicating with one LC at a time in a loop fashion at a selected time interval, e.g., 10 s. Each LC responds to CC's request by providing its own data. After all LCs respond, the combined data is fetched from CC by the HEM through a data request. The system is designed such that the HEM gets data from all LCs at 1-min intervals.

If the HEM decides a status change for any of the loads, it issues an ON/OFF command to the CC. The message includes which LC (or fan) to change status and requested status (ON or OFF). The CC forwards the command to appropriate LC by its own ON/OFF command sent to the LC. Upon receiving this command, the LC changes relay status, and replies with a response to CC. Once CC receives this response (or not), it replies to the HEM with a success (or failure) confirmation response.

Message Frame Formats Used in Communication

All communication uses a strictly defined message frame format. This format ensures packet integrity and proper authentication between communicating parties. The format used in this system with an example frame is given in Table 1.

The start delimiter marks the beginning of a new message frame. It is immediately followed by the message length for marking the end of the frame. The next two fields denote the source and the function of the message, respectively. The source identifies where the message is coming from (CC, LC or HEM) and the function helps identify whether it is a data request/response or a control command/response. Next comes the actual payload which contains the data, command or responses. It is followed by a checksum byte. The start and end delimiters, length and checksum help ensure integrity and proper transfer of message frames.

The example frame shown in Table 1 shows an HEM ON command for LC1. The source byte 0x48 means the HEM unit, and the function byte 0x34 means ON/OFF command. In the command field, the first byte 0x30 indicates which LC (in this case: first one) and the next byte indicates ON (0x31) or OFF (0x30). It is then followed by the end delimiter 0x0D and the checksum byte. The message length (5 bytes) is sent as the second byte of the frame after the start delimiter (0xEE). The same format is used for other messages in the system.

Software Design of Load Controller and Monitoring/Control Systems

This section describes the design of embedded software for CC and LCs and the monitoring/controlling system interface

Table 1 Message frame format for communication

Start delimiter	Message length	Source	Function	Data/command	Checksum
0xEE	0x05	0x48	0x34	0x30310D	0x15

Table 2 List of major tasks/flags and their functions in software

Task/flag	Function
Process_Opmode	Main thread of CC Algorithm
Process_UART	Processes received UART message
Process_SPI_CC	Processes received SPI message in CC
SPIFRAMERECEIVED	Indicate if a SPI frame has been received
DoTask1_Send_Data_HEM	Flag to trigger sending data to HEM
CC_Send_Data_Response	Sends data to HEM from CC
DoTask1_Send_LC_C4_x	Flag to trigger sending command to LC x
DoTask1_Send_LC_C1_x	Flag to trigger sending data req to LC x
Fan_auto	Flag for Fan auto/manual mode
Check_All_Temp	Checks LC and Ambient temperature
Get_int_temp	Receive data from ambient sensor
TX_ACTIVE	Flag indicating ongoing SPI transmission
UARTFRAMERECEIVING	Flag indicating ongoing UART reception

(GUI) at the HEM unit. Table 2 lists major tasks/flags and their functions in the embedded CC and LC code, which will aid in discussing embedded software designs of CC and LCs.

Embedded Software Design of CC

The CC algorithm is based on an interrupt and flag type mechanism. The main process of the algorithm keeps checking for different flag status changes in the code and takes action whenever a flag is set by an interrupt function or other processes in the code. Once the action is complete, it resets the flag and again keeps tracking flag changes. The interrupt functions run as separate threads which are invoked by interrupt events, like UART/SPI byte reception or timer interval. These interrupt functions set appropriate flags for the main process to respond to an event. As the embedded program starts, the algorithm first initializes different sub-systems and ports. It then enables the global interrupt and goes into an infinite loop that calls the main operational process, i.e., Process_Opmode(), again and again. The following subsections discuss key threads of the CC software.

Process_Opmode

The Process_Opmode() is the main thread of the algorithm that keeps tracking flag status changes and invokes actions accordingly. Figure 5 shows the flow chart of Process_Opmode().

In Process_Opmode(), first it checks for received and unprocessed UART messages. If there is one or more unread messages, then it reads the next unread message from buffer and processes it through Process_UART() function. After processing the message, it clears the buffer and decrements UART_unread_messages. Once there is no unread message left and no action in the UART interrupt function, pointers to UART message buffers are reset to the first buffer.

Next it checks for the frame received in the SPI buffer. If there is a frame received, it processes the frame through Process_SPI_CC(), then clears the buffer and the flag. Then it checks for different task flags. If the Send_Data_HEM flag has been set, then it sends the data response to HEM through UART. If a C4 flag for any LC has been set indicating ON/OFF command, it invokes the CC_Send_ONOFF_Command_check() function which takes care of the ON/OFF command task. Similarly, if a C1 flag has been set indicating data request, it calls CC_Send_Data_Request_check() function which takes care of the data request task. Finally, if the fan is in auto mode and the Check_temp_status flag has been set, it calls Check_All_Temp() function which checks all temperature values and decides if the box fan needs to be turned ON or OFF. For all task related functions, the flag is reset once the task is complete.

Process_UART

This function is called when there is an unread UART frame in the buffer. It checks the message in UART_RX_buffer for start delimiter, length and checksum consistencies. Then, it checks the address and command in the message. If it is a data request from the HEM, it triggers the DOTASK1_SEND_DATA_HEM flag bit, so that other part of the program sends data response to the HEM. If an ON/OFF command from HEM is found in the message, it first checks if the ON/OFF command is for the fan or LC. If it is for the fan, the fan is appropriately turned ON/OFF and a response is sent back to the HEM. If the ON/OFF command is for an LC, it calls a function that checks which LCs to change status, and sets appropriate C4 flag bits.

Process_SPI_CC

This function is called when a SPI frame is received. This function processes the SPI_RX_buffer. It first checks for message start delimiter, length, and checksum verification. If these parameters are acceptable, the message address and command are checked. For data response from an LC, it updates the data into the table. For ON/OFF acknowledgement from an LC, it updates the device status and sends response back to the HEM.

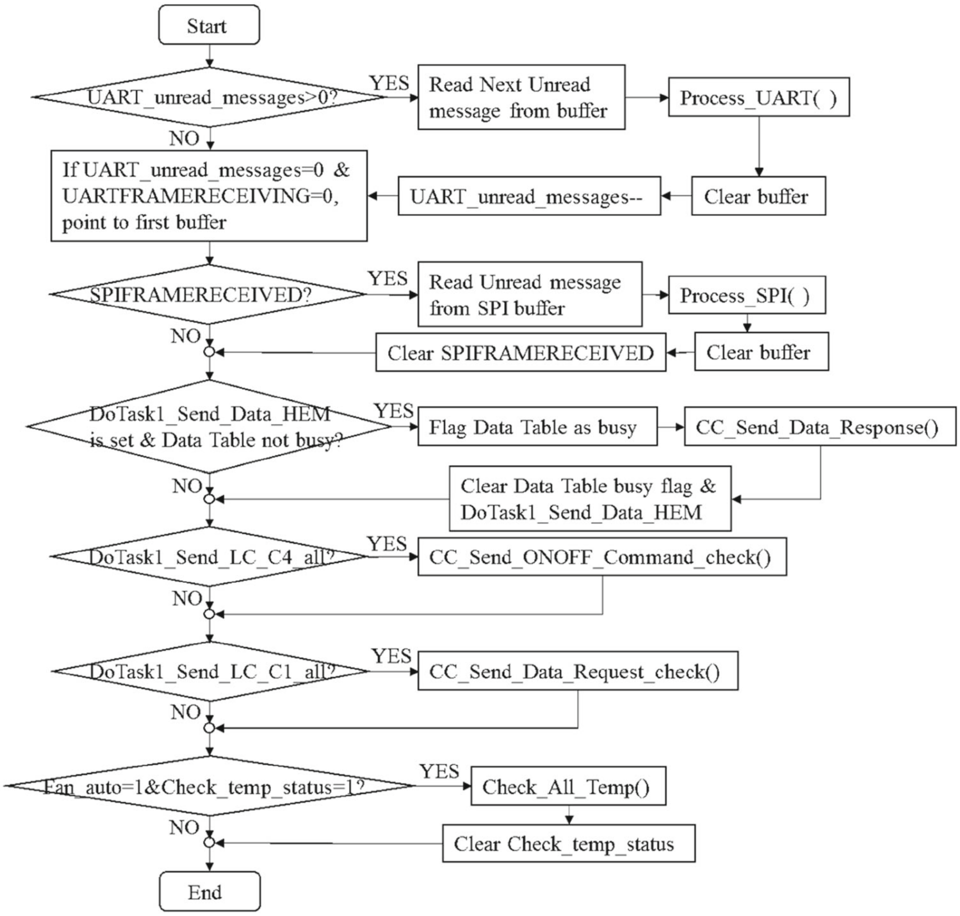
Start
UART_unread_messages>0?
YES
Read Next Unread message from buffer
Process_UART( )
NO
If UART_unread_messages=0 & UARTFRAMERECEIVING=0, point to first buffer
UART_unread_messages--
Clear buffer
SPIFRAMERECEIVED?
YES
Read Unread message from SPI buffer
Process_SPI( )
NO
Clear SPIFRAMERECEIVED
Clear buffer
DoTask1_Send_Data_HEM is set & Data Table not busy?
YES
Flag Data Table as busy
CC_Send_Data_Response()
NO
Clear Data Table busy flag & DoTask1_Send_Data_HEM
DoTask1_Send_LC_C4_all?
YES
CC_Send_ONOFF_Command_check()
NO
DoTask1_Send_LC_C1_all?
YES
CC_Send_Data_Request_check()
NO
Fan_auto=1&Check_temp_status=1?
YES
Check_All_Temp()
NO
Clear Check_temp_status
End

Fig. 5 Flow chart of Process_Opmode

Timer Interrupt Handler

This interrupt counts the duration in seconds on each call. At every LC update interval (e.g., 60 s), it periodically sets the C1 flags in DoTask1 flag register. This is to allow other parts of the program to periodically send data requests to LCs. The timer second counter is reset after a pre-set interval. Also at this interval, the ambient temperature data is fetched and stored, and the Check_temp_status flag is set.

SPI Receive Interrupt Handler

This interrupt is triggered whenever a byte has been received through SPI. If SPI transmission is going on, indicated by TX_Active bit in SPI_status flag register, the received byte is ignored. Otherwise the received byte is put in the SPI_RX_buffer. The routine also checks for the message format. If a complete message has been received in the buffer, it sets the SPIFRAMERECEIVED flag bit.

UART Receive Interrupt Handler

It is similar in operation as SPI receive interrupt handler. As UART can either transmit or receive at a time, the transmission active checking is not necessary for UART. Hence, the routine receives bytes and puts them in the UART_RX_buffer. While the frame is being received, a flag is kept set so that buffer pointers are not altered. When a complete frame has been received, it increments UART_unread_messages to indicate a new message has been received and sets the pointer to the next buffer for future UART frames.

Embedded Software Design of LC

The LC algorithm has two main operations. First, it uses the ADC to accumulate power data from the load, store them and provide them to the CC if requested. Second, it controls the relay connecting the load to power and changes its status based on ON/OFF command from the CC. The algorithm first initializes different sub-systems and enables the global interrupt. Then, it goes into an infinite loop calling the WaitLPM() repeatedly.

WaitLPM

The WaitLPM process is the main thread of the algorithm which processes the ADC data and SPI received frames by

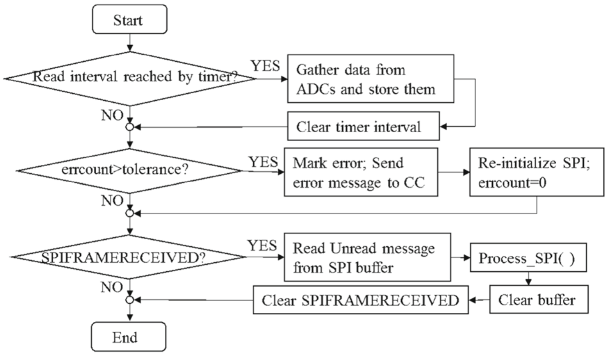
Start
Read interval reached by timer?
YES
Gather data from ADCs and store them
NO
Clear timer interval
errcount>tolerance?
YES
Mark error; Send error message to CC
Re-initialize SPI; errcount=0
NO
SPIFRAMERECEIVED?
YES
Read Unread message from SPI buffer
Process_SPI( )
NO
Clear SPIFRAMERECEIVED
Clear buffer
End

Fig. 6 Flow chart of WaitLPM

calling other functions according to flags. At a preset interval, it gathers the data from ADCs and stores them in a buffer. It then checks for the errcount value. If it exceeds a tolerance value, this indicates a communication error during the last SPI frame reception/transmission. In that case, the program marks the communication error and sends an error message to the CC. Then, it re-initializes the SPI port to prevent further communication errors and resets errcount. Finally WaitLPM checks for SPIFRAMERECEIVED, and if a frame has been received, it calls Process_SPI_LC() to process the message, then clears the buffer and resets the flag. Figure 6 summarizes this process.

Process_SPI_LC

This function is called when a SPI frame has been received. This function processes the SPI_RX_buffer. It first checks for the message start delimiter, length, and checksum verification. If these parameters are acceptable, the message address and command are checked. For data request message from CC, it packages the last acquired data into message format and puts it into the writing buffer which is used by the SPI interrupt handler routine to dispatch the message to the CC. In case of an ON/OFF command, it updates the relay status according to the command and then puts the response message to the CC in the writing buffer. In both cases, it flags SPI_TX_ACTIVE so that SPI interrupt handler knows that it is ready to transmit.

SPI Receive Interrupt Handler

This interrupt is triggered whenever a byte has been received through SPI. If SPI is in its transmitting stage, indicated by SPI_TX_ACTIVE, then it keeps putting the bytes from writing buffer into the SPI TX buffer for the CC to fetch them until the end of the message frame. Once the full message has been dispatched, it clears the SPI_TX_ACTIVE flag and resets errcount. If SPI is not in its transmitting stage, the received byte is put in the SPI_RX_buffer and errcount is incremented. If a complete message has been received in the buffer, it sets the SPIFRAMERECEIVED flag bit. Otherwise, errcount is kept incrementing, which indicates that there has been some form of communication errors in the SPI.

Software Design of HEM

The HEM algorithm used with the proposed load controller system is similar to [7]. A discussion of the algorithm is presented here for the convenience of the readers. In this HEM algorithm, four power-intensive loads in a house are controlled to meet the demand restriction imposed by a utility during a peak demand period. An incentive based DR program is assumed here where the customer gets a one-time statement credit within a contract with the utility to respond to its DR signals. The DR contains the DR event duration (hours) and the demand limit (kW) to be maintained during the event. The HEM also has pre-set load priority and preference settings from the customer. The following loads can be controlled by the proposed load controller system:

Space Heater (SH)

A space heater is a resistive load that can be turned ON or OFF by an LC. The customer specifies the preferred indoor temperature range to HEM. This can be in the form of a specified temperature set-point $T_{SH,s}$ and deadband ΔT_{SH}. The status of space heating load will be ON if the room temperature is below $(T_{SH,s} - \Delta T_{SH})$ and OFF if it is above $(T_{SH,s} + \Delta T_{SH})$. It will continue the previous state otherwise.

Water Heater (WH)

A water heater is also a resistive load and can be controlled based on the customer preferred hot water temperature range. The customer specifies a hot water temperature set-point $T_{WH,s}$ and deadband ΔT_{WH}. Like SH, the status is decided based on the hot water temperature.

Clothes Dryer (CD)

In the proposed algorithm, only the heating circuit of a clothes dryer is controlled by an LC, leaving the operation of the motor circuit unchanged. The customer specifies the maximum OFF/minimum ON durations and the job completion time. Clothes dryer will be turned ON if it has been OFF for more than the maximum OFF time. Similarly, once ON, it can be turned OFF after the minimum ON time. In any case, the required total time for CD will be scheduled before the job completion time.

Electric Vehicle Charger (EV)

An LC can be used to control both the charging signal to an EV charger as well as the power supply for the charging. The customer specifies their preference for minimum charging duration before a cut-off and the charge completion time. If the EV charging is ON, it will not be turned off within the minimum charging duration. The charging will be turned OFF once the state of charge (SOC) reaches the full charge value. The required total time for finishing the EV charging will be scheduled before the charge completion time.

A customer also specifies a priority ranking for each of these loads. An example of priority and preference settings for the four loads are given in Table 3 for clarity.

Based on the pre-set load priorities, the HEM algorithm controls four loads to keep the total household demand below the specified demand limit during the DR event, while maintaining customer's preference settings. The algorithm is summarized in Fig. 7. The algorithm first gathers all required information, including real time status and power consumption information from load controllers, temperature data from sensors and the DR event signal from an electric utility. Based on these inputs, at each time step (1-min intervals in this case), the HEM checks for preference setting violation for each load and demand limit violation. If there is a violation, the highest priority load(s) will be served first and the lowest priority load(s) will be turned OFF to keep the total household power demand within the demand limit. In all cases, the algorithm is guaranteed to maintain the total household consumption below the demand limit during the DR event, while maintaining preference settings based on the pre-set load priority. Detailed simulation results were presented in [7] to show the efficacy of the algorithm. The algorithm was developed as software with user interface that runs on a PC, representing an HEM unit.

Table 3 Sample load priority and preference settings in a house

Load	Load priority	Customer preference
Water heater	1	Hot water temperature: 115°F (±5°F)
Space heater	2	Room temperature: 76°F (±2°F)
Clothes dryer	3	Finish job by midnight Maximum OFF time: 30 mins Minimum ON time: 30 mins
EV charger	4	Finish charging by 8AM Minimum ON time: 30 mins

Implementation Results

The proposed load controller system was developed with CC and LC hardware boards in a packaged box and the embedded software was developed as described in previous sections. The implementation results from this system are discussed in the following sub-sections.

A Test System in the Laboratory Environment

A test system was set up in the laboratory environment. Due to unavailability of actual household loads in the laboratory setting, four resistive heater loads were used. Each heater was connected to one LC. Their ratings were 208 V/240 V AC 1.3/1.5 A.

Figure 8 shows logged data of 12 h for the four loads connected to the four LCs in the load controller system. The data was collected with a sampling rate of one measurement per minute for each LC, as outlined previously in the details of the design. The voltage, current, power and temperature measurements for the four LCs have been shown in the figure. As can be seen, during the 12 h log, no data drop or bad data was recorded, confirming stable communication performance. The temperature measurements of the boards as well

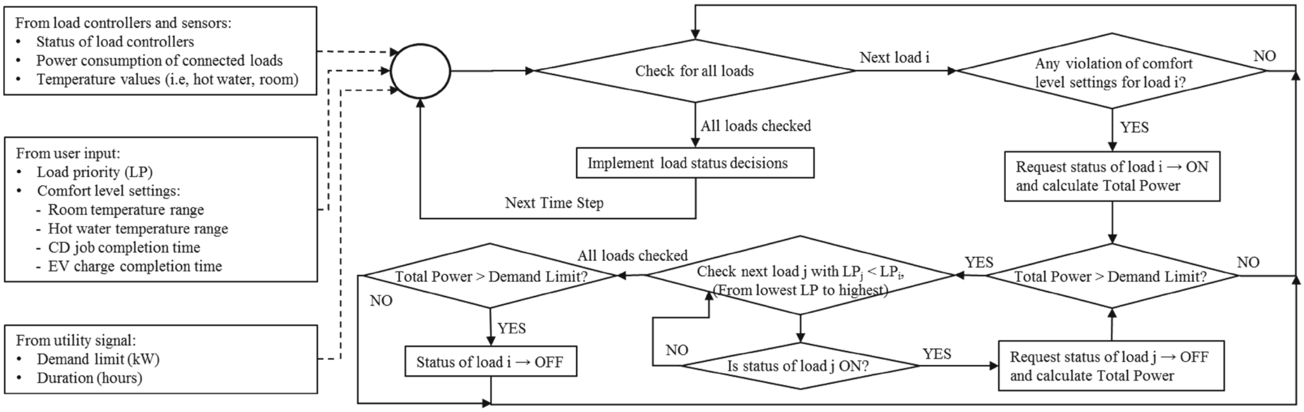
From load controllers and sensors:
• Status of load controllers
• Power consumption of connected loads
• Temperature values (i.e, hot water, room)
From user input:
• Load priority (LP)
• Comfort level settings:
- Room temperature range
- Hot water temperature range
- CD job completion time
- EV charge completion time
From utility signal:
• Demand limit (kW)
• Duration (hours)
Check for all loads
Next load i
Any violation of comfort level settings for load i?
NO
YES
All loads checked
Implement load status decisions
Request status of load i → ON and calculate Total Power
Next Time Step
All loads checked
Total Power > Demand Limit?
NO
YES
Status of load i → OFF
Check next load j with LP$_j$ < LP$_i$, (From lowest LP to highest)
YES
Total Power > Demand Limit?
NO
NO
Is status of load j ON?
YES
Request status of load j → OFF and calculate Total Power

Fig. 7 Flow Chart of HEM algorithm used with the proposed load controller system

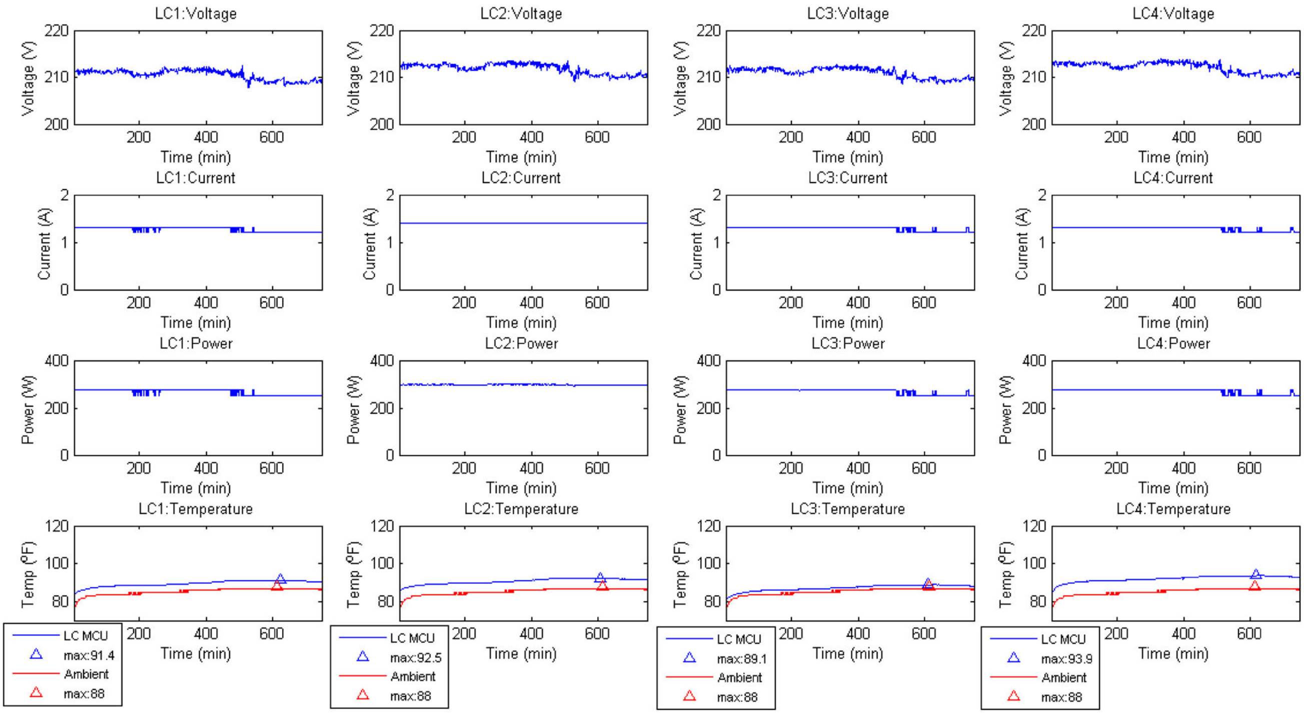

Fig. 8 Logged voltage, current, power, MCU temperature and ambient temperature data for four LCs for 12 h

as the box are also stable, which shows safe operation for long duration.

Case Study Analysis with HEM

The analysis of continuous-time control with a discrete DR event was performed to showcase the applicability of the proposed load controller system.

System Setup and Assumptions

The test system described in the previous sub-section was used to simulate the control of a hypothetical average single family house of 2500 sq. ft. In this setup, the four heater loads connected to the four LCs are used to represent the four power-intensive household loads: Water Heater (WH), Space heater (SH), Clothes Dryer (CD) and Electric Vehicle charger (EV). The power consumption of the representative loads are scaled up to match the power consumption of actual loads. The scaling factors are given in Table 4.

Table 4 Scaling factors for heater loads to represent loads in a house

Represented Load	Scaling factor
Water heater	16
Space heater	8
Clothes dryer	14
EV charger	11.5

The load priorities and comfort preferences are set according to Table 3. Simulated values are used for room temperature and hot water temperature input. The other loads in the household that are not controlled by the HEM are referred to as critical loads and their total consumption is taken from the RELOAD database [30]. A computer with the previously described HEM software is used to represent the HEM unit. It has a ZigBee module to send control commands to the proposed load controller based on the power measurements (with scaling factors), priority and preference settings, DR scenario and the HEM algorithm, as described in “Software Design of HEM” section.

Case Scenarios

To analyze the effectiveness of the continuous-time control of the load controller system and its responsiveness in discrete-event control based on a pre-defined DR scenario, two case studies were performed with the HEM system and the load controller system. In case 1, a system with no DR event was considered. In this case, the HEM was allowed to control the loads based on their preference setting violation, and no demand restriction was considered. In the second case, a DR event was considered to be scheduled from 5 pm to 7 pm. During this period, a demand limit of 8kW was considered to be imposed on the system, and the HEM algorithm was used to make decisions of load control based on load priority

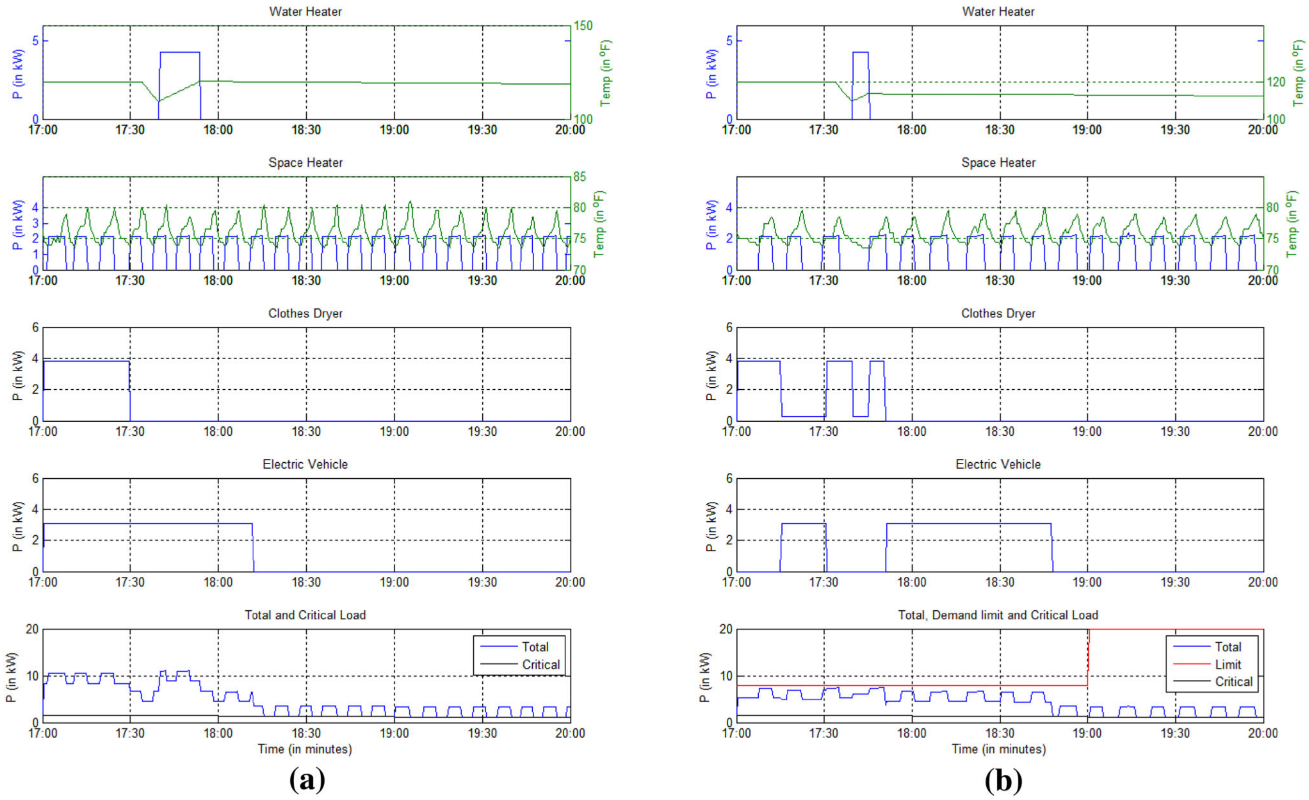

Fig. 9 Demonstration of HEM operated load controller system with scaled power consumption of represented loads from 5 pm to 8 pm: **a** without a DR event, **b** with a DR event

and preference settings to maintain the total consumption within the demand limit. Figure 9 shows results of these two cases, presenting the power consumption of four loads and the total consumption for a side-by-side comparison.

Case 1: Without a DR event The consumption profiles of four loads and the total and critical load consumption of the house from 5pm to 8pm without DR is shown in Fig. 9a. Also, for water heater and space heater, the hot water temperature and room temperature are plotted in the same sub plots. As can be seen from the figure, loads were operated according to their preference settings to maintain the temperature values and job completion times. No demand restriction was employed during the whole 3-h experiment, and a peak load of 11 kW was observed at around 5:40 pm.

Case 2: With a DR event The consumption profiles for this case are shown in Fig. 9b. In this case, the demand limit of 8 kW was imposed from 5 pm to 7 pm. Due to the demand limit, the HEM algorithm reschedules load operations based on load priorities to maintain the total consumption. For example, around 5:40 pm, when a peak load was observed in the previous case, in this case, the WH was allowed to run due to its higher priority, and both SH and CD operations were delayed to keep the total consumption within limit. A similar decision was implemented around 5:50 pm, when SH and CD were allowed to run, but the EV operation was postponed. As can be seen from Fig. 9b, during the DR event, the HEM system successfully restricted the total household consumption within the specified 8 kW limit.

As can be seen from the case study analysis, the load controller system displayed stable operation for both continuous-time and discrete-event control. As the HEM algorithm makes load scheduling decisions based on actual consumption and demand limit restrictions, it is guaranteed to have convergence in decision-making, by delaying operation of low-priority loads. The load controller system performs properly according to the HEM commands, as demonstrated in the case study analysis.

Stability Against Communication Delays/Errors

Control loops in a network-based control system can be affected by delays or packet drops in the sensor network. Therefore, the stability and robustness of the control scheme has to be carefully designed based on the probability of occurrences of such delays/errors.

Packet Delays in a ZigBee Network-Based Control System

There are prevalent studies on how sensor networks are affected by packet delays.

Authors in [9] discuss different wireless communication technologies for residential energy management applications

Table 5 Average round-trip time delay between HEM and CC

Distance between HEM unit and CC	Time delay (ms)
1 meter	240
10 meter	265

and provide experiment results from a ZigBee based HEM implementation. In that paper, they show that the packet delivery ratio, delay and jitter of the ZigBee based sensor network improve as the packet size is decreased. Results show that a packet size of 32 bytes yields a delivery ratio of almost 90 %, with average end-to-end delay of 0.75 seconds and a jitter of 0.04 seconds.

Authors in [31] also evaluated serial ZigBee devices for application in wireless networked control systems and they observed an average of 14 ms round-trip delay for 2 bytes of data transfer.

These results were considered while designing the packets for our proposed load controller system. As discussed in the "Message Frame Formats Used in Communication" section, a control command packet in this scheme consists of only 8 bytes. A data response packet from CC to HEM is the largest packet in this scheme, and it still uses only 51 bytes. All of the designed packets include start and end delimiters, length and checksum bytes to ensure integrity of the packet.

From our own experiments based on the designed packet frames for the load controller, measured average round-trip time delays of 60 sample-sets of package transfer between the HEM and the CC for a distance of 1 meter and 10 meters [32] are around 250 ms, as given in Table 5.

Dynamics of a Networked Control System Under Time Delays

The load controller system proposed in this paper is a networked control system which comprises of four control units and their related sensor units. If we consider that the system's input stimulus causes an exponential decay response on the sensors' outputs and also bring in the time delays between the control inputs and system outputs into consideration, a multi-input multi-output (MIMO) model of the networked control system can be proposed.

To simplify the formulation without loss of generality, a networked control system of two control units and two sensors is considered. Denoting the system inputs as u's, system outputs as y's, steady state gains as k's, time constants as τ's and time delays as T's, the s-domain representation can be given as:

$$\begin{aligned} y_1(s) &= \frac{k_{11}e^{-T_{11}s}}{\tau_{11}s+1}u_1(s) + \frac{k_{12}e^{-T_{12}s}}{\tau_{12}s+1}u_2(s) \\ y_2(s) &= \frac{k_{21}e^{-T_{21}s}}{\tau_{21}s+1}u_1(s) + \frac{k_{22}e^{-T_{22}s}}{\tau_{22}s+1}u_2(s) \end{aligned} \tag{1}$$

Reformulating Eq. (1), one can obtain:

$$\begin{aligned} &(\tau_{11}s+1)(\tau_{12}s+1)\,y_1(s)\,e^{(T_{11}+T_{12})s} \\ &\quad = k_{11}(\tau_{12}s+1)e^{T_{12}s}u_1(s) + k_{12}(\tau_{11}s+1)\,e^{T_{11}s}u_2(s) \end{aligned} \tag{2}$$

And,

$$\begin{aligned} &(\tau_{21}s+1)(\tau_{22}s+1)\,y_2(s)\,e^{(T_{21}+T_{22})s} \\ &\quad = k_{21}(\tau_{22}s+1)\,e^{T_{22}s}u_1(s) + k_{22}(\tau_{21}s+1)\,e^{T_{21}s}u_2(s) \end{aligned} \tag{3}$$

Using Taylor series expansion, the exponential terms can be simplified: $e^{(T_{11}+T_{12})s} \simeq 1+(T_{11}+T_{12})\,s$; $e^{T_{12}s} \simeq 1+T_{12}s$; $e^{T_{11}s} \simeq 1+T_{11}s$

Similarly: $e^{(T_{21}+T_{22})s} \simeq 1+(T_{21}+T_{22})\,s; e^{T_{22}s} \simeq 1+T_{22}s$; $e^{T_{21}s} \simeq 1+T_{21}s$

Substituting the Taylor series expansions into Eqs. (2) and (3),

$$\begin{aligned} &(\tau_{11}s+1)(\tau_{12}s+1)\left[1+(T_{11}+T_{12})\,s\right]y_1(s) \\ &\quad = k_{11}(\tau_{12}s+1)(1+T_{12}s)\,u_1(s) \\ &\qquad + k_{12}(\tau_{11}s+1)(1+T_{11}s)\,u_2(s) \end{aligned} \tag{4}$$

$$\begin{aligned} &(\tau_{21}s+1)(\tau_{22}s+1)\left[1+(T_{21}+T_{22})\,s\right]y_2(s) \\ &\quad = k_{21}(\tau_{22}s+1)(1+T_{22}s)\,u_1(s) \\ &\qquad + k_{22}(\tau_{21}s+1)(1+T_{21}s)\,u_2(s) \end{aligned} \tag{5}$$

After intermediate operation and reordering of terms in Eq. (4):

$$\begin{aligned} &[\tau_{11}\tau_{12}(T_{11}+T_{12})\,s^3 \\ &\qquad + \{\tau_{11}\tau_{12}+(\tau_{11}+\tau_{12})(T_{11}+T_{12})\}\,s^2 \\ &\qquad + \{(\tau_{11}+\tau_{12})+(T_{11}+T_{12})\}\,s+1]y_1(s) \\ &\quad = k_{11}\left[\tau_{12}T_{12}s^2+(\tau_{12}+T_{12})\,s+1\right]u_1(s) \\ &\qquad + k_{12}\left[\tau_{11}T_{11}s^2+(\tau_{11}+T_{11})\,s+1\right]u_2(s)\,\cdot \end{aligned} \tag{6}$$

Similarly, from Eq. (5):

$$\begin{aligned} &[\tau_{21}\tau_{22}(T_{21}+T_{22})\,s^3 \\ &\qquad + \{\tau_{21}\tau_{22}+(\tau_{21}+\tau_{22})(T_{21}+T_{22})\}\,s^2 \\ &\qquad + \{(\tau_{21}+\tau_{22})+(T_{21}+T_{22})\}\,s+1]y_2(s) \\ &\quad = k_{21}\left[\tau_{22}T_{22}s^2+(\tau_{22}+T_{22})\,s+1\right]u_1(s) \\ &\qquad + k_{22}\left[\tau_{21}T_{21}s^2+(\tau_{21}+T_{21})\,s+1\right]u_2(s) \end{aligned} \tag{7}$$

To further simplify, the following coefficients are defined for Eq. (6):

$$a_{1,1} = \tau_{11}\tau_{12}\left(T_{11} + T_{12}\right);$$
$$a_{2,1} = \left\{\tau_{11}\tau_{12} + \left(\tau_{11} + \tau_{12}\right)\left(T_{11} + T_{12}\right)\right\};$$
$$a_{3,1} = \left\{\left(\tau_{11} + \tau_{12}\right) + \left(T_{11} + T_{12}\right)\right\}; \quad a_{4,1} = 1.$$
$$b_{1,1} = k_{11}\tau_{12}T_{12}; \quad b_{2,1} = k_{11}\left(\tau_{12} + T_{12}\right); \quad b_{3,1} = k_{11};$$
$$b_{4,1} = k_{12}\tau_{11}T_{11}; \quad b_{5,1} = k_{12}\left(\tau_{11} + T_{11}\right); \quad b_{6,1} = k_{12}.$$

Similarly, coefficients are defined for Eq. (7):

$$a_{1,2} = \tau_{21}\tau_{22}\left(T_{21} + T_{22}\right);$$
$$a_{2,2} = \left\{\tau_{21}\tau_{22} + \left(\tau_{21} + \tau_{22}\right)\left(T_{21} + T_{22}\right)\right\};$$
$$a_{3,2} = \left\{\left(\tau_{21} + \tau_{22}\right) + \left(T_{21} + T_{22}\right)\right\}; \quad a_{4,2} = 1.$$
$$b_{1,2} = k_{21}\tau_{22}T_{22}; \quad b_{2,2} = k_{21}\left(\tau_{22} + T_{22}\right); \quad b_{3,2} = k_{21};$$
$$b_{4,2} = k_{22}\tau_{21}T_{21}; \quad b_{5,2} = k_{22}\left(\tau_{21} + T_{21}\right); \quad b_{6,2} = k_{22}.$$

Using the notation of coefficients, the dynamics of the networked control system in Eqs. (6) and (7) can be written as:

$$\begin{aligned} a_{1,1}y_1^{(3)} &+ a_{2,1}\ddot{y}_1 + a_{3,1}\dot{y}_1 + a_{4,1}y_1 \\ &= b_{1,1}\ddot{u}_1 + b_{2,1}\dot{u}_1 + b_{3,1}u_1 + b_{4,1}\ddot{u}_2 \\ &\quad + b_{5,1}\dot{u}_2 + b_{6,1}u_2 \end{aligned} \tag{8}$$

$$\begin{aligned} a_{1,2}y_2^{(3)} &+ a_{2,2}\ddot{y}_2 + a_{3,2}\dot{y}_2 + a_{4,2}y_2 \\ &= b_{1,2}\ddot{u}_1 + b_{2,2}\dot{u}_1 + b_{3,2}u_1 + b_{4,2}\ddot{u}_2 \\ &\quad + b_{5,2}\dot{u}_2 + b_{6,2}u_2 \end{aligned} \tag{9}$$

Equivalently, the description about the dynamics of the networked control system in Eq. (8) can be written as:

$$\begin{aligned} y_1^{(3)} = &-\frac{a_{2,1}}{a_{1,1}}\ddot{y}_1 - \frac{a_{3,1}}{a_{1,1}}\dot{y}_1 - \frac{a_{4,1}}{a_{1,1}}y_1 \\ &+ \frac{b_{1,1}}{a_{1,1}}\ddot{u}_1 + \frac{b_{2,1}}{a_{1,1}}\dot{u}_1 + \frac{b_{3,1}}{a_{1,1}}u_1 \\ &+ \frac{b_{4,1}}{a_{1,1}}\ddot{u}_2 + \frac{b_{5,1}}{a_{1,1}}\dot{u}_2 + \frac{b_{6,1}}{a_{1,1}}u_2 \end{aligned} \tag{10}$$

Similarly, the description about the dynamics of the networked control system in Eq. (9) can be written as:

$$\begin{aligned} y_2^{(3)} = &-\frac{a_{2,2}}{a_{1,2}}\ddot{y}_2 - \frac{a_{3,2}}{a_{1,2}}\dot{y}_2 - \frac{a_{4,2}}{a_{1,2}}y_2 \\ &+ \frac{b_{1,2}}{a_{1,2}}\ddot{u}_1 + \frac{b_{2,2}}{a_{1,2}}\dot{u}_1 + \frac{b_{3,2}}{a_{1,2}}u_1 \\ &+ \frac{b_{4,2}}{a_{1,2}}\ddot{u}_1 + \frac{b_{5,2}}{a_{1,2}}\dot{u}_2 + \frac{b_{6,2}}{a_{1,2}}u_2 \end{aligned} \tag{11}$$

Now, the state variables can be defined as following:

$$z_1 = y_1; \ z_2 = \dot{y}_1; \ z_3 = \ddot{y}_1; \ z_4 = y_2; \ z_5 = \dot{y}_2; \ z_6 = \ddot{y}_2$$

Also, the control variables and their derivatives can be considered as additional state variables:

$$z_7 = u_1; \ z_8 = \dot{u}_1; \ z_9 = u_2; \ z_{10} = \dot{u}_2$$

The following variables are defined as new control inputs:

$$v_1 = \ddot{u}_1; \ v_2 = \ddot{u}_2$$

And, the following functions are defined:

$$\begin{aligned} f_1 = &-\frac{a_{2,1}}{a_{1,1}}z_3 - \frac{a_{3,1}}{a_{1,1}}z_2 - \frac{a_{4,1}}{a_{1,1}}z_1 \\ &+ \frac{b_{2,1}}{a_{1,1}}z_8 + \frac{b_{3,1}}{a_{1,1}}z_7 + \frac{b_{5,1}}{a_{1,1}}z_{10} + \frac{b_{6,1}}{a_{1,1}}z_9 \end{aligned} \tag{12}$$

$$g_{11} = \frac{b_{1,1}}{a_{1,1}}; \quad g_{12} = \frac{b_{4,1}}{a_{1,1}} \tag{13}$$

$$\begin{aligned} f_2 = &-\frac{a_{2,2}}{a_{1,2}}z_6 - \frac{a_{3,2}}{a_{1,2}}z_5 - \frac{a_{4,2}}{a_{1,2}}z_4 \\ &+ \frac{b_{2,2}}{a_{1,2}}z_8 + \frac{b_{3,2}}{a_{1,2}}z_7 + \frac{b_{5,2}}{a_{1,2}}z_{10} + \frac{b_{6,2}}{a_{1,2}}z_9 \end{aligned} \tag{14}$$

$$g_{21} = \frac{b_{1,2}}{a_{1,2}}; \ g_{22} = \frac{b_{4,2}}{a_{1,2}} \tag{15}$$

Then, the dynamics of the networked control system under time delays can be written as:

$$\begin{aligned} \dot{z}_3 &= f_1 + g_{11}v_1 + g_{12}v_2 \\ \dot{z}_6 &= f_2 + g_{21}v_1 + g_{22}v_2 \end{aligned} \tag{16}$$

The same can also be written in a matrix form:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \\ \dot{z}_7 \\ \dot{z}_8 \\ \dot{z}_9 \\ \dot{z}_{10} \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ f_1 \\ z_5 \\ z_6 \\ f_2 \\ z_8 \\ 0 \\ z_{10} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ g_{11} & g_{12} \\ 0 & 0 \\ 0 & 0 \\ g_{21} & g_{22} \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{17}$$

Compensation of Delay in a Networked Control System

Designing controllers for networked control system based on continuous time control theory results in unsatisfactory performance [33]. Therefore, to handle the network delay effects, controllers are designed in discrete time. Examples can be found in [31] and [33], where discrete time PID controllers with filtering on derivative part and anti-windup of integrative part were used. Kalman filtering methods are also popular in this case [34–38]. Authors in [34] use derivative-free non-linear Kalman filtering to deal with input disturbances in control of doubly fed induction generators, whereas author in [35] uses derivative-free non-linear Kalman filtering for distributed state-estimation under communication delays and packet drops. Authors in [36] model the arrival of measurement observations as a random process and use discrete Kalman filtering formulation to show that beyond a critical value for arrival rate of the observations, an unbounded state error covariance occurs. Authors in [37] use data fusion process on measurements that are time-correlated using Kalman filters. Authors in [38] utilize the differential flatness property of the system to design a control approach to compensate for delays.

Most of the approaches to deal with the delays in networked control system, determine the discrete-time period of measurement/control based on the average delay times and their respective effect on the control performance. In our proposed load controller system, the control application uses 1-min interval for issuing or changing control commands to the LCs. This 1-min interval was selected based on the requirement of the demand response application as well as the observation of average packet delays. As shown in the previous sub-section, the delays are in millisecond range, therefore the 1-min period provides ample time to collect sensor data even with maximum possible delays.

Stability Against Communication Errors

Besides communication delays, corrupted packets and message drops are also possible. For time critical control loops where retransmission is not feasible, use of Kalman Filters to estimate missing/corrupted sensor data is useful. But, in the proposed control scheme in this paper, the 1-min interval allows sufficient time to attempt retransmission of data. Therefore, a selective repeat (SR) automatic repeat request (ARQ) scheme [8] is employed to ensure that a successful communication is made within the 1-min interval. This also keeps the control algorithms less computation intensive for deployment in low-cost controllers. In the worst case scenario, where successive re-transmission attempts fail and an updated data cannot be obtained by the HEM in a given minute, the last obtained correct data is used for decision making. As can be seen from the load profiles of the residential loads to be controlled by the proposed load controller system [8], the assumption that power consumption remains almost same for a load in the same state for successive minutes is valid for the concerning loads. Also, as the state change is only issued by the HEM itself, the reuse of past data in case of missing data does not cause any incorrect decision making. The experimental results shown in the previous section verify this.

To summarize, the overall communication and control algorithm is carefully designed and experimentally verified to be robust against communication issues in this network based control system.

Conclusion

This paper presents a microcontroller and ZigBee based load controller system for monitoring and controlling power intensive non-communicating residential loads. It discusses the design of the system, covering details of hardware and software design and communication schemes. It also presents implementation results. The proposed design uses wired and reliable SPI communication between the CC and load controllers (LCs). This also reduces the communication burden of LCs, thereby allowing use of low cost microcontrollers. The compact and robust design allows for ease of installation and usage and prevents the need of expensive electrical re-wiring/outlet installations or even more expensive replacement of old loads with modern loads.

The proposed design can be customized with the use of different communication schemes (such as WiFi, Z-Wave) with appropriate changes in the hardware and protocol design. Any other HEM application can also be used, provided that the target loads are compatible with the control of relays in the LCs and the HEM unit follows the appropriate protocol to communicate with the CC. Ratings of the relays can also be modified, together with the driving circuit, if there is a need to make the LCs compatible with loads of different voltage and current ratings.

Overall, the proposed design can serve as a low cost and reliable solution for enabling control of traditional loads in any HEM application. This will encourage customer participation in a residential demand response program in the smart grid environment.

The future work can include the use of different communication technologies and protocols and comparing the impact of different communication technologies on the design and performance of the system.

Acknowledgements This work was supported in part by U.S. National Science Foundation under Grant IIP-1114314.

References

1. Albadi, M., El-Saadany, E.: Demand response in electricity markets: an overview. In: Power Engineering Society General Meeting, pp. 1–5 (2007)
2. Piette, M.A., Watson, D., Motegi, N., Kiliccote, S., Linkugel, E.: Automated demand response strategies and commissioning commercial building controls. In: Proceedings of the 14th National Conference on Building Commissioning, San Fransisco (April 2006)
3. Li, J., Chuang, J.Y., Xiao, J., Hong, J.W., Boutaba, R.: On the design and implementation of a home energy management system.

In: Proceeding of the 6th International Symposium on Wireless and Pervasive Computing (ISWPC), pp. 1–6, 23–25 Feb 2011

4. Han, J., Choi, C.S., Park, W.K., Lee, I.: Green home energy management system through comparison of energy usage between the same kinds of home appliances. In: Proceeding of the 15th IEEE International Symposium on Consumer Electronics (ISCE), pp. 1–4 (2011)
5. Pedrasa, M.A.A., Spooner, T.D., MacGill, I.F.: Coordinated scheduling of residential distributed energy resources to optimize smart house energy services. IEEE Trans. Smart Grid **1**(3), 320–331 (2010)
6. Mohsenian-Rad, A.H., Leon-Garcia, A.: Optimal residential load control with price prediction in real-time electricity pricing environments. IEEE Trans. Smart Grid **1**(2), 120–133 (2010)
7. Pipattanasomporn, M., Kuzlu, M., Rahman, S.: An algorithm for intelligent home energy management and demand response analysis. IEEE Trans. Smart Grid **3**(4), 2166–2173 (2012)
8. Saha, A., Kuzlu, M., Pipattanasomporn, M., Rahman, S., Elma, O., Selamogullari, S., Uzunoglu, M., Yagcitekin, B.: A robust building energy management algorithm validated in a smart house environment. Intell. Ind. Syst. **1**(2), 163–174 (2015)
9. Erol-Kantarci, M., Mouftah, H.T.: Wireless sensor networks for cost efficient residential energy management in the smart grid. IEEE Trans. Smart Grid **2**(2), 314–325 (2011)
10. Han, D., Lim, J.: Design and implementation of smart house energy management systems based on ZigBee. IEEE Trans. Consum. Electron. **56**(3), 1417–1425 (2010)
11. Son, Y.S., Pulkkinen, T., Moon, K.Y., Kim, C.: Home energy management system based on power line communication. IEEE Trans. Consum. Electron. **56**(3), 1380–1386 (2010)
12. Han, J., Choi, C.-S., Park, W.-K., Lee, I., Kim, S.-H.: Smart home energy management system including renewable energy based on ZigBee and PLC. IEEE Trans. Consum. Electron. **60**(2), 198–202 (2014)
13. Han, J.-S., Choi, C.-S., Lee, I.-W.: More efficient home energy management system based on ZigBee communication and infrared remote controls. IEEE Trans. Consum. Electron. **57**(1), 85–89 (2011)
14. Huang, H.-Y., Yen, J.-Y., Chen, S.-L., Ou, F.-C.: Development of an intelligent energy management network for building automation. IEEE Trans. Autom. Sci. Eng. **1**(1), 14–25 (2004)
15. Byun, J., Hong, I., Park, S.: Intelligent cloud home energy management system using household appliance priority based scheduling based on prediction of renewable energy capability. IEEE Trans. Consum. Electron. **58**(4), 1194–1201 (2012)
16. Hu, Q., Li, F.: Hardware design of smart home energy management system with dynamic price response. IEEE Trans. Smart Grid **4**(4), 1878–1887 (2013)
17. Gabbar, H.A., Eldessouky, A.S.: Energy semantic network for building energy management. Intell. Ind. Syst. **1**(3), 213–231 (2015)
18. Schneider Electric, Wiser Home Management, https://www.wiserenergy.com/
19. General Electric, GE Energy Management, http://www.geenergymanagement.com/
20. Honeywell, Energy Management products, http://honeywell.com/Solutions-Technologies/Pages/energy.aspx#Management
21. Digi International, Digi ZBee Smart Plug, http://www.digi.com/products/wireless-modems-peripherals/wireless-range-extenders-peripherals/xbee-smart-plug-zb
22. EnTek International LLC, Remote Appliance Controller, http://entekllc.com/wp-content/uploads/2011/02/mc140-datasheet-1-26-11.pdf
23. Ecobee, Smart Plug, http://www.ecobee.com/wp-content/themes/ecobee/etc/smartPlug_a4_april9.pdf
24. Belkin, WeMo Switch, http://www.belkin.com/us/p/P-F7C027/
25. Dominion Virginia Power, Smart Cooling Rewards Program. https://www.dom.com/residential/dominion-virginia-power/ways-to-save/energy-conservation-programs/smart-cooling-rewards
26. Rappahannock Electric Cooperative, Water Heater Control. https://www.myrec.coop/res/save-energy/water-heater-control.cfm
27. ZigBee Alliance: ZigBee smart energy profile specification, Version 1.1 (2011)
28. Gungor, V.C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., Hancke, G.P.: Smart Grid technologies: communication technologies and standards. IEEE Trans. Ind. Inform. **7**(4), 529–539 (2011)
29. Saha, A., Kuzlu, M., Pipattanasomporn, M.: Demonstration of a home energy management system with smart thermostat control. In: Proceedings of the Innovative Smart Grid Technologies (ISGT) Conference, pp. 1–6 (2013)
30. RELOAD Database Documentation and Evaluation and Use in NEMS. http://www.onlocationinc.com/LoadShapes-Reload2011.pdf
31. Godoy, E.P., Scorzoni, F., Porto, A.J.V.: Evaluating serial ZigBee devices for application in wireless networked control systems, IECON 2012—38th Annual Conference on IEEE Industrial Electronics Society, Montreal, pp. 3176–3181 (2012)
32. Kuzlu, M., Pipattanasomporn, M., Rahman, S.: Hardware demonstration of a home energy management system for demand response applications. IEEE Trans. Smart Grid **3**(4), 1704–1711 (2012)
33. Eriksson, L.: PID controller design and tuning in networked control systems, Doctorate Thesis, Department of Automation and Systems Technology, Helsinki University of Technology, Finland (2008)
34. Rigatos, G., Siano, P., Zervos, N., Cecati, C.: Control and disturbances compensation for doubly fed induction generators using the derivative-free nonlinear kalman filter. IEEE Trans. Power Electron. **30**(10), 5532–5547 (2015)
35. Rigatos, G.G.: Derivative-free distributed filtering for MIMO robotic systems under delays and packet drops. Int. J. Adv. Robot. Syst. **10**, 133 (2013)
36. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M.I., Sastry, S.S.: Kalman filtering with intermittent observations. IEEE Trans. Autom. Control **49**(9), 1453–1464 (2004)
37. Langlois, D., Elliott, J., Croft, E.A.: Sensor uncertainty management for an encapsulated logical device architecture. Part II: a control policy for sensor uncertainty. In: American Control Conference, 2001. Proceedings of the 2001, Arlington, vol. 6, pp. 4288–4293 (2001)
38. Rigatos, G., Siano, P., Rahman, S., Busawon, K., Binns, R., Zervos, N.: A flatness-based control approach to networked control under sensor delays, ISI Internal Research Report (2016)

Avijit Saha (S'06—IEEE) received his B.Sc. degree in Electrical and Electronic Engineering (EEE) from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 2009. He served as a Lecturer at the department of EEE in Ahsanullah University of Science and Technology, Dhaka, Bangladesh from March 2010 to July 2011. Currently he is working towards his Ph.D. degree in Electrical Engineering in Virginia Tech. His research interests include the areas of smart grid, demand response, energy management systems, communication technologies and protocols, software architecture, embedded systems, internet of things and cyber security.

Murat Kuzlu (M'11, SM'14—IEEE) joined Virginia Tech's Department of Electrical and Computer Engineering as a post-doctoral fellow in 2011. He received his B.Sc., M.Sc., and Ph.D. degrees in Electronics and Telecommunications Engineering from Kocaeli University, Turkey, in 2001, 2004, and 2010, respectively. From 2005 to 2006, he worked as a Global Network Product Support Engineer at the Nortel Networks, Turkey. In 2006, he joined the Energy Institute of TUBITAK-MAM, where he worked as a senior researcher at the Power Electronic Technologies Department. His research interests include smart grid, demand response, smart metering systems, wireless communication and embedded systems.

Manisa Pipattanasomporn (S'01, M'06, SM'11—IEEE) joined Virginia Tech's Department of Electrical and Computer Engineering as an assistant professor in 2006. She manages multiple research grants from the U.S. National Science Foundation, the U.S. Department of Defense and the U.S. Department of Energy, on research topics related to smart grid, microgrid, energy efficiency, load control, renewable energy and electric vehicles. She received her Ph.D. in electrical engineering from Virginia Tech in 2005. Her research interests include renewable energy systems, energy efficiency, distributed energy resources, and the smart grid.

Saifur Rahman (S'75, M'78, SM'83, F'98, LF'15—IEEE) received the Ph.D. degree in electrical engineering from Virginia Tech, Blacksburg, VA, USA in 1978. He is the founding director of the Advanced Research Institute, Virginia Tech, Arlington, VA, USA, where he is the Joseph R. Loring Professor of Electrical and Computer Engineering. He is also the director of the Center for Energy and the Global Environment at Virginia Tech. Prof. Rahman is the founding Editor-in-Chief of the IEEE Electrification Magazine and the IEEE Transactions on Sustainable Energy. He is the IEEE PES President-elect for 2016 and 2017, and served as a Vice President of PES from 2009 to 2013. He is currently serving as a member of the Board of Governors of the IEEE Society on Social Implications of Technology. In 2006, he served on the IEEE Board of Directors as the Vice President for Publications. He served as the Chair of the U.S. National Science Foundation (NSF) Advisory Committee for International Science and Engineering from 2010 to 2013. He is a recipient of the IEEE Millennium Medal.